

1 IRELL & MANELLA LLP
Morgan Chu (SBN 70446)
2 (mchu@irell.com)
Jonathan S. Kagan (SBN 166039)
3 (jkagan@irell.com)
1800 Avenue of the Stars, Suite 900
4 Los Angeles, California 90067-4276
5 Telephone: (310) 277-1010
Facsimile: (310) 203-7199
6
7 David C. McPhie (SBN 231520)
(dmcphie@irell.com)
8 Douglas J. Dixon (SBN 275389)
(ddixon@irell.com)
9 Nima Hefazi (SBN 272816)
(nhefazi@irell.com)
10 840 Newport Center Drive, Suite 400
Newport Beach, California 92660-6324
11 Telephone: (949) 760-0991
12 Facsimile: (949) 760-5200
13 Attorneys for Defendant
JUNIPER NETWORKS, INC.
14

15 UNITED STATES DISTRICT COURT
16 FOR THE NORTHERN DISTRICT OF CALIFORNIA
17 SAN FRANCISCO DIVISION

18 IMPLICIT NETWORKS, INC.,
19 Plaintiff,
20 v.
21 JUNIPER NETWORKS, INC.,
22 Defendant.
23
24
25

Case No. C 10-4234 SI

**JUNIPER NETWORKS, INC.'S NOTICE
OF MOTION AND MOTION FOR
SUMMARY JUDGMENT OF NON-
INFRINGEMENT**

Date: December 14, 2012
Time: 9:00 a.m.
Courtroom: 10

NOTICE OF MOTION

PLEASE TAKE NOTICE that on Friday, December 14, 2012, at 9:00 a.m., or as soon thereafter as counsel may be heard, pursuant to Rule 56 of the Federal Rules of Civil Procedure, defendant Juniper Networks, Inc. ("Juniper") will, and hereby does, move for an order granting summary judgment of non-infringement of claims 1, 15, and 35 of U.S. Patent No. 6,629,163 ("the '163 patent") and claims 1, 4, and 10 of U.S. Patent No. 7,711,857 ("the '857 patent") (collectively, the "asserted claims"). Juniper's motion is based upon this Notice, the Memorandum of Points and Authorities below, the supporting Expert Declaration of Dr. Peter Alexander ("Alexander Declaration") and the supporting documents attached thereto, the Declaration of Nima Hefazi in Support of Juniper Networks, Inc.'s Motion for Summary Judgment of Non-Infringement, the supporting documents attached thereto, any subsequently filed briefs, the pleadings and papers filed in this action, and any other arguments, evidence, and matters submitted to the Court, at the hearing or otherwise.

Dated: November 9, 2012

Respectfully submitted,

IRELL & MANELLA LLP

By: /s/ Nima Hefazi

David C. McPhie (dmcphe@irell.com)
 Douglas J. Dixon (ddixon@irell.com)
 840 Newport Center Drive, Suite 400
 Newport Beach, California 92660-6324
 Telephone: (949) 760-0991
 Facsimile: (949) 760-5200

Morgan Chu (mchu@irell.com)
 Jonathan S. Kagan (jkagan@irell.com)
 1800 Avenue of the Stars, Suite 900
 Los Angeles, California 90067-4276
 Telephone: (310) 277-1010
 Facsimile: (310) 203-7199

*Attorneys for Defendant
 Juniper Networks, Inc.*

TABLE OF CONTENTS

	Page
TABLE OF AUTHORITIES	iii
STATEMENT OF ISSUES TO BE DECIDED	1
STATEMENT OF RELEVANT FACTS	2
A. Technical Background and the Patents-in-Suit	2
B. Procedural History	3
1. Implicit Initially Serves Broad Infringement Contentions Including the Juniper “Multiservices” Products	3
2. Implicit Invests Significant Resources in Analyzing the Juniper Multiservices Software Code	3
3. Implicit Decides to Drop its Infringement Claims Against the Multiservices Products But Still Relies on the Multiservices Code Analysis to Accuse Other Juniper Products	4
4. After Reading Juniper’s Non-Infringement Expert Report, Implicit’s Expert Backtracks on His Software Code Opinions	5
ARGUMENT	7
I. LEGAL PRINCIPLES	7
II. IMPLICIT’S INFRINGEMENT CLAIMS FAIL BECAUSE THEY RELY ON THE WRONG EVIDENCE AND DO NOT PINPOINT WHERE THE ELEMENTS OF THE PATENTS-IN-SUIT ARE FOUND IN THE ACCUSED PRODUCTS	8
A. Implicit’s Infringement Expert Improperly Relies on Source Code for a Juniper Product Not Accused of Infringement in this Case	8
B. Implicit Cannot Show That the Multiservices Module is an Appropriate Proxy for Analyzing Infringement by the SRX and J Series Products	9
C. Implicit’s Infringement Case Cannot Stand Without the Source Code “Analysis” in the Nettles Report	13
III. IN THE ALTERNATIVE, IMPLICIT’S INFRINGEMENT CLAIMS FAIL EVEN IF ALL OF DR. NETTLES’S CITED EVIDENCE IS DEEMED APPLICABLE TO THE ACCUSED PRODUCTS, AS HE STILL FAILS TO SATISFY AT LEAST THREE ELEMENTS OF THE ASSERTED CLAIMS	16
A. Implicit Cannot Show That The Accused Products Satisfy The “Selecting Individual Components” Limitation	17

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Page

B. Implicit Cannot Show That The Accused Products Include A
“Plurality of Components” Satisfying The Claims, Including The
“State Information” Limitations 19

C. Implicit Cannot Show That The Accused Products Perform The Step
Of “Dynamically Identifying . . . After The First Packet.” 22

IV. IMPLICIT HAS FAILED TO DEMONSTRATE ANY ACTUAL
INFRINGEMENT IN THE ACCUSED MANNER OR
CONFIGURATION. 25

CONCLUSION 25

TABLE OF AUTHORITIES

Page(s)

Cases

<i>Alloc, Inc. v. Pergo, L.L.C.</i> , 2010 WL 3860382 (E.D. Wis. Sept. 30, 2010)	11
<i>Anderson v. Liberty Lobby, Inc.</i> , 477 U.S. 242 (1986)	7
<i>Arthur A. Collins, Inc. v. N. Telecom, Ltd.</i> , 216 F.3d 1032 (Fed. Cir. 2000)	8
<i>Atlantic Thermoplastics Co. v. Faytex Corp.</i> , 974 F.2d 1299 (Fed. Cir. 1994)	18
<i>Big Baboon Corp. v. Dell, Inc.</i> , 723 F. Supp. 2d 1224 (C.D. Cal. 2010)	13
<i>Celotex Corp. v. Catrett</i> , 477 U.S. 317 (1986)	7
<i>Eugene Baratto, Textures, LLC v. Brushstrokes Fine Art, Inc.</i> , 701 F.Supp.2d 1068 (W.D. Wis. March 24, 2010)	10, 11, 13
<i>Exigent Tech., Inc. v. Atrana Solutions, Inc.</i> , 442 F.3d 1301 (Fed. Cir. 2006)	7
<i>Fujitsu Ltd. v. LG Electronics, Inc.</i> , 620 F.3d, 1321 (Fed. Cir. 2010)	25
<i>GE v. Joiner</i> , 522 U.S. 136 (1997)	11
<i>In re Oracle</i> , 2009 U.S. Dist. LEXIS 50995 (N.D. Cal. June 16, 2009)	16, 21
<i>L&W, Inc. v. Shertech, Inc.</i> , 471 F.3d 1311 (Fed. Cir. 2006)	11
<i>Medtronics Vascular, Inc. v. Boston Scientific Corp.</i> , 2008 U.S. Dist. LEXIS 53373 (E.D. Tex. July 11, 2008)	11
<i>Novartis v. Ben Venue</i> , 271 F.3d 1043 (Fed. Cir. Nov. 7, 2001)	7, 15
<i>Pixion, Inc. v. Citrix Systems, Inc.</i> , 2012 WL 3313533 (N.D. Cal. Aug. 13, 2012)	15, 16, 17, 21
<i>Rambus, Inc. v. Hynix Semiconductor, Inc.</i> , 642 F. Supp. 2d 970 (N.D. Cal. Nov. 24, 2008)	8
<i>Regents of Univ. of Cal. v. Dako N. Am., Inc.</i> , 2009 WL 1083446 (N.D. Cal. Apr. 22, 2009)	7

1		Page(s)
2		
3	<i>Scanner Techs. Corp. v. ICOS Vision Sys. Corp.</i> ,	
4	528 F.3d 1365 (Fed. Cir. 2008)	14
5	<i>Taylor v. List</i> ,	
6	880 F.2d 1040 (9th Cir. 1989)	7
7	<u>Other Authorities</u>	
8	Fed. R. Civ. P. 30(b)(6)	19
9	Fed. R. Civ. P. 56	7
10	Fed. R. Evid. 402	8
11	Fed. R. Evid. 702	8
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		

STATEMENT OF ISSUES TO BE DECIDED

Implicit bears the burden to establish that each of the accused Juniper products practices each and every limitation of the asserted claims in this case. Based on the undisputed factual record in this case, it cannot do so. Therefore, Juniper seeks to dispose of each of Implicit's causes of action in this motion for summary judgment.

Implicit accuses several of Juniper's computer networking products of infringing its patents. Juniper made the source code for these products available to Implicit and its experts for inspection, and Implicit spent months poring through this code. When it came time for Implicit—through its expert—to explain which portions of Juniper's code corresponded with the various limitations of its patent, however, it could not do so. Rather, Implicit cites only to source code for products that it has already dismissed from this case. Thus, Implicit has presented no relevant source code evidence relating to the products that are currently in-suit. (*See* Part II, below)

Moreover, even the source code Implicit cites for the now-dismissed products fails to establish infringement because it does not address each limitation in the asserted claims, a basic requirement for an infringement finding. Specifically, Implicit presents no evidence that the source code from these now-dismissed products preforms any of the following limitations:

1. "selecting individual components" so that formats are "compatible";
2. "plurality of components" including "state information" steps; and
3. "dynamically identifying." (*See* Part III, below)

Furthermore, Dr. Nettles fails to provide any proof of actual infringement in the specific manner he claims falls within the scope of the claims. (*See* Part IV, below.)

It is well-established that, by the time a patent case reaches the summary judgment stage, a patentee with the burden of proving infringement is obligated to present evidence (including expert evidence) adequate to demonstrate that there is a triable issue of fact with respect to the basic elements of its infringement claims. Quite simply, Implicit has failed to do this.

Accordingly, Juniper submits that the Court should grant summary judgment of non-infringement for the reasons set forth in detail below.

STATEMENT OF RELEVANT FACTS

A. Technical Background and the Patents-in-Suit

Computer networks, such as the Internet, send and receive messages by breaking those messages into a number of different “packets,” sending those packets over a network, and then reassembling the packets into the message at the destination. Implicit claims that the patents-in-suit cover a particular method of handling multi-packet messages. The specific method claimed in Implicit’s patents require the use of several different software routines or “components” (each of which performs a single step) to help process the packets once they reach their destination. The asserted claims further require that the system examine the first packet in a sequence in order to figure out how best to process all packets in the message. Because this processing involves multiple steps, and each component performs only one step, it is necessary to select and use multiple software routines or “components” (like modular stations on an assembly line). This selection of components is then stored so it can be used in processing each subsequent packet that is part of the same message.

The claims also require that the components in the sequence do certain things. For example, claim 1 of the ‘163 patent requires that each component convert data in packets from an input to an output format. And each of the asserted claims requires each component retrieve, use, and store certain information called “state information” relating to the processing of a particular packet, so that information can be used “when processing the next packet of the message.”

The Court’s Claim Construction Order provides further guidance regarding the claimed limitations of “selecting individual components to create [or form] . . . [a non-predefined] sequence of components.” First, the *timing* of this limitation is important—the claims and Court’s claim construction provide that the “sequence of components” that the packets of a message travel through can only be identified “*after the first packet is received.*” See Hefazi Ex. 1 (Claim Construction Order) at 7:13-15.¹ As the patent explains, this is necessary to avoid the “overhead” of otherwise having to provide “each possible series of conversion routines” in advance. Second,

¹ “Hefazi Ex. ___” refers to the exhibits to the Declaration of Nima Hefazi in Support of Juniper’s Motion for Summary Judgment of Non-Infringement, executed November 9, 2012.

the claims impose a necessary constraint of ensuring *compatibility* between components in a sequence. In other words, as the system passes packets through each of the “sequence” of components, it must determine that the *output format* of each component *matches the input format* of the next component in line. *See Id.* at 11:6-10.

B. Procedural History

1. *Implicit Initially Serves Broad Infringement Contentions Including the Juniper “Multiservices” Products*

In May 2011, Implicit’s served its first set of Patent L.R. 3-1 contentions on Juniper with broad infringement allegations against virtually every single product Juniper makes, including (among others): (1) SRX series products, (2) J series products, and (3) Juniper routers (M, MX, T, TX series) when used with an optional add-on card called the “Multiservices” module. Hefazi Ex. 2 at 7–9. These infringement contentions consisted almost entirely of block quotes from public materials relating to these various Juniper products. *See, e.g.,* Hefazi Ex. 3.

2. *Implicit Invests Significant Resources in Analyzing the Juniper Multiservices Software Code*

Implicit demanded that Juniper produce software source code² for all of the accused products, and Juniper agreed to do so. The source code (which was massive) was made available on a secured computer at the offices of Juniper’s outside counsel. Counsel for Implicit hired a software consultant, Pavel Treskunov, to review the code, and Mr. Treskunov spent literally months in the offices of Juniper’s counsel doing so.

Mr. Treskunov prepared a written report detailing the results of his source code review, which he shared with Implicit (but not yet with Juniper). *See* Ex. 4. Although the report was entitled “JUNOS Analysis,” it focused entirely on the operation of the Juniper services routers installed with the Multiservices module (or “MS module”), described as “optionally installable and hot swappable hardware” used in Juniper routers. *Id.* at 5. The report went on to describe the operation of the Multiservices module—and in particular the software known as the “Multiservices daemon”—using flow charts and discussion of specific code passages. *Id.* at 8–24.

² “Source code” is “a way that humans give instructions to computers,” in a way that can be used to show other computer programmers what those instructions are. Hefazi Ex. 5 (10/9/2012 Nettles Depo. Tr.) at 46:20 – 25.

1 After Implicit had the opportunity to review Juniper's source code, the parties engaged in
 2 some discovery motion practice regarding Implicit's failure to provide code specific references to
 3 the portions of the source code that allegedly corresponded to the elements of the asserted claims.
 4 This Court then ordered Implicit to disclose source code pincites supporting its infringement
 5 claims on an element-by-element basis. Dkt. Nos. 71 & 84. On June 7, 2012, Implicit served
 6 amended infringement contentions with some code citations taken from the Treskunov report,
 7 which Implicit affirmed fully complied with the Court's orders. In these amended contentions,
 8 Implicit also dropped its infringement allegations against all Juniper products other than Juniper's
 9 routers with the Multiservices module, and the SRX and J series products.

10 3. *Implicit Decides to Drop its Infringement Claims Against the Multiservices*
 11 *Products But Still Relies on the Multiservices Code Analysis to Accuse*
Other Juniper Products

12 On August 14, 2012, Implicit deposed Todd Regonini, Juniper's Vice President of Systems
 13 Engineering. During this deposition, Mr. Regonini explained that Juniper's largest customers
 14 rarely use the Multiservices modules in the particular mode that Implicit had accused of
 15 infringement. Hefazi Ex. 6 at 103:15 – 104:5. Shortly after this deposition (and apparently based
 16 on this testimony), Implicit dropped all infringement claims based on the Multiservices products.

17 By this time, Implicit had retained its testifying expert on the issue of infringement,
 18 Dr. Scott Nettles. For reasons unknown to Juniper, Dr. Nettles elected to not perform his own
 19 review of the Juniper source code. Hefazi Ex. 5 (10/9/2012 Nettles Depo. Tr.) at 100:4-11,
 20 105:11-13. Instead, Dr. Nettles relied only on a copy of Mr. Treskunov's report, which only
 21 addressed the now non-accused Multiservices module. Hefazi Ex. 7 (10/19/2012 Nettles Depo.
 22 Tr.) at 165:7-21. Dr. Nettles never met with or spoke with Mr. Treskunov, nor did he even ask
 23 any questions regarding the analysis in Mr. Treskunov's report. Hefazi Ex. 5 at 104:17 – 105:7;
 24 *id.* at 282:13 – 283:3.

25 The problem Dr. Nettles faced, therefore, was that his infringement report needed to
 26 address the SRX and J series products but Mr. Treskunov's analysis of source code was for an
 27 entirely different line of products—products using the Multiservices module. And Mr. Treskenov
 28 had not even tried to confirm that the currently-accused products operate in the same way; he was

1 unable to confirm that “all of the Juniper accused products operated in essentially the same
2 manner as the Multiservices module for purposes of infringement”—or, if he had, he had “never
3 communicated” any such thing to Dr. Nettles. Hefazi Ex. 7 at 169:6-17.

4 With little time remaining before the expert report deadline, Implicit devised the following
5 “solution” to its problem: Dr. Nettles would continue to rely heavily on Mr. Treskunov’s analysis,
6 but he would try to obscure the fact that the source code that formed the core of his infringement
7 allegations was not used by the accused products. Indeed, large portions of Dr. Nettles’s report
8 were simply flow charts he cut and pasted into the technical section of his own (Dr. Nettles’s)
9 expert report. Hefazi Ex. 5 at 280:9-281:11. In so doing, Dr. Nettles was careful to *omit* the
10 *context* of Mr. Terskunov’s analysis, which explained that it had been directed solely to the
11 structure and operation of the Multiservices module.³ In place of this crucial contextual
12 information, Dr. Nettles substituted a few paragraphs implying (but not stating) that Juniper’s
13 products were all essentially the same, because they employed code from a single, “integrated”
14 operating system known as “JUNOS.” Hefazi Ex. 8 (Nettles Report, Appx. A) ¶¶ 7-9. The only
15 support cited for this proposition was an excerpt from a Juniper marketing document. *Id.*

16 4. *After Reading Juniper’s Non-Infringement Expert Report, Implicit’s Expert*
17 *Backtracks on His Software Code Opinions*

18 After receiving a copy of the Nettles report, Juniper asked its technical expert Dr. Peter
19 Alexander to review and prepare a rebuttal report. Dr. Alexander performed an exhaustive
20 analysis of Dr. Nettles’s infringement allegations and the evidence cited in his report. As an initial
21 matter, Dr. Alexander observed that there was “disconnect” between the alleged supporting
22 evidence that Dr. Nettles had cited in his report and the Implicit allegations of infringement as to
23 specific Juniper products, because Dr. Nettles had “rel[ied] on documents and source code
24 pertaining to products that are not even accused of infringement in this case.” *See, e.g.,* Alexander
25 Ex. A ¶¶ 71, 75-79.⁴ As Dr. Alexander explained:

26 Fundamental to [Implicit’s] obligation [to prove infringement] is the

27 ³ Dr. Nettles later testified that this omission had been “intentional” because he thought it
28 was “not important to the infringement analysis at all.” Ex. 7 at 175:11 – 176:22.

⁴ “Alexander Ex. ___” refers to the exhibits to the Declaration of Peter Alexander in Support
of Juniper’s Motion for Summary Judgment of Non-Infringement, executed November 9, 2012.

necessity of providing some basis to believe that the source code and other evidence allegedly supporting Dr. Nettles's infringement analysis actually pertain to the accused Juniper products. Having failed to satisfy this basic threshold requirement, Dr. Nettles's opinions regarding infringement necessary fail.

Id. ¶ 79. Moreover, although he was under no obligation to do so (because Implicit bears the burden of proving infringement), Dr. Alexander reviewed some portions of the Juniper source code relating to the SRX and J series products that Dr. Nettles had not cited or reviewed, which further supported his conclusion that Dr. Nettles had relied on the wrong code. *Id.* ¶¶ 80-86.

Finally, in the interest of providing a complete report, Dr. Alexander analyzed the cited "evidence" of infringement assuming that the source code Dr. Nettles relied upon was actually used by the accused products (which, of course, it is not). As a result of this analysis, Dr. Alexander concluded that the code and other evidence Dr. Nettles had cited still failed to satisfy numerous elements of the asserted claims in his infringement analysis. For example, Dr. Alexander found that Dr. Nettles had failed to show that the accused SRX and J series products met the claim limitations of (1) "selecting individual components," (2) a "plurality of components" (including the detailed "state information" requirements), and (3) "dynamically identifying." Accordingly, Dr. Alexander concluded that, even accepting as true Dr. Nettles's mistaken assumptions regarding the applicability of the cited code and other evidence, Implicit still failed to carry its burden that the SRX and J series products infringed the patents-in-suit.

Dr. Alexander's report explained these (and other) problems with Implicit's infringement theories. Perhaps not surprisingly, after reviewing this report, Dr. Nettles essentially conceded that the code on which he based his opinion did not apply to the accused products:

Q. Is it your opinion, sitting here today that the source code you cite in your report is used for each and every one of the accused products . . . ?

A. . . . Your expert's report draws that into question. And so I think at this point I don't have – I think I would – would need to do further investigation to have a firm opinion about the answer to this question

See Hefazi Ex. 5 at 244:5-24; *see also* Hefazi Ex. 7 at 180:24 – 183:1. Dr. Nettles confirmed that because he was "not as confident about that . . . I'm certainly not going to testify that I'm

absolutely sure that every one of these pieces of code that are cited applies to the SRX or the J series.” *Id.* at 182:18-22.⁵

ARGUMENT

I. LEGAL PRINCIPLES

Summary judgment is proper where “there is no genuine issue as to any material fact and . . . the moving party is entitled to judgment as a matter of law.” Fed. R. Civ. P. 56(c). A dispute involving a material fact is genuine only “if the evidence is such that a reasonable jury could return a verdict for the nonmoving party.” *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 248 (1986). The moving party “bears the initial burden of informing the court of the basis for its motion.” *Celotex Corp. v. Catrett*, 477 U.S. 317, 323 (1986). The moving party has no burden to negate or disprove matters on which the non-moving party will have the burden of proof at trial. *Id.* at 325. “On an issue for which the opposing party will have the burden of proof at trial, the moving party need only point out ‘that there is an absence of evidence to support the nonmoving party’s case.’” *Regents of Univ. of Cal. v. Dako N. Am., Inc.*, 2009 WL 1083446, at *5 (N.D. Cal. Apr. 22, 2009). To meet this initial burden, “[n]othing more is required than the filing of a summary judgment motion stating that the patentee had no evidence of infringement and pointing to the specific ways in which accused systems did not meet the claim limitations.” *Exigent Tech., Inc. v. Atrana Solutions, Inc.*, 442 F.3d 1301, 1309 (Fed. Cir. 2006).

Once the moving party meets this initial burden, “the opposing party cannot rest on the mere allegation or denials of a pleading, but must “go beyond the pleadings and by [the party’s] own affidavits, or by the ‘depositions, answers to interrogatories, and admissions on file’ designate ‘specific facts showing that there is a genuine issue of for trial.’” *Celotex*, 477 U.S. at 324. The opposing party also may not rely solely on conclusory allegations unsupported by factual data. *Taylor v. List*, 880 F.2d 1040, 1045 (9th Cir. 1989); *Novartis v. Ben Venue*, 271 F.3d 1043, 1054-55 (Fed. Cir. 2001) (affirming summary judgment of non-infringement in absence of

⁵ It was also during the Nettles deposition that Juniper learned about the existence of the Treskunov report (including Dr. Nettles’s reliance on that report). Hefazi Ex. 5 at 105:8-20, 280:9-281:23. Juniper then requested and received (belatedly) a copy of this report and other papers that Dr. Nettles had received from Mr. Treskunov and considered in forming his opinions.

1 “theoretical and factual foundation” for infringement allegations).

2 **II. IMPLICIT’S INFRINGEMENT CLAIMS FAIL BECAUSE THEY RELY ON THE**
 3 **WRONG EVIDENCE AND DO NOT PINPOINT WHERE THE ELEMENTS OF**
 4 **THE PATENTS-IN-SUIT ARE FOUND IN THE ACCUSED PRODUCTS**

5 **A. Implicit’s Infringement Expert Improperly Relies on Source Code for a**
 6 **Juniper Product Not Accused of Infringement in this Case**

7 It is axiomatic that a plaintiff claiming infringement by a given accused product must
 8 support that claim with evidence actually corresponding to that product. This basic prerequisite is
 9 reflected, for example, in Rule 402 of the Federal Rules of Evidence (“[i]rrelevant evidence is not
 10 admissible”) as well as Rule 702 (expert opinions must be “reliable” and “help the trier of fact”).
 11 It is Implicit, as the party asserting infringement, that “bears the burden of persuasion at trial as to
 12 whether or not each of [Juniper’s] *accused products* infringe its claims.” *Rambus, Inc. v. Hynix*
 13 *Semiconductor, Inc.*, 642 F. Supp. 2d 970, 976 (N.D. Cal. Nov. 24, 2008) (emphasis added).
 14 Thus, in order to avoid summary judgment on this issue, Implicit’s expert on the issue of
 15 infringement “must set forth the factual foundation for his opinion. . . in sufficient detail for the
 16 court to determine whether the factual foundation would support a finding of infringement” as to
 17 the accused products. *Arthur A. Collins, Inc. v. N. Telecom, Ltd.*, 216 F.3d 1032, 1047-48 (Fed.
 18 Cir. 2000).

19 Implicit cannot satisfy this burden because its infringement claims are not supported by
 20 requisite evidence pertaining to the accused SRX and J series products. Specifically, Implicit’s
 21 expert Dr. Nettles has failed to provide a proper “factual foundation” for his infringement opinions
 22 in reliance on the cited Multiservices source code. His deposition testimony on this point is clear:

23 Q. The Multiservices module is not used on the SRX and J series
 24 products, correct?

25 A. Yes, sir, that’s correct.

26 Q. And you knew that at the time that you prepared your expert
 27 report regarding infringement in this case, correct?

28 A. Yes, sir, I don’t think there’s any representation in my report that
 it is used in those products.

Hefazi Ex. 7 at 187:1-9. In fact, Dr. Nettles elected to ignore this crucial fact; his report does not
 cite to any evidence that the source code he analyzed (or anything similar) is implemented in the

1 accused products.⁶ He testified only that he would “need to do further investigation to have a firm
2 opinion” about whether this was the case, Hefazi Ex. 5 at 244:5-24, and there is no evidence here
3 of any such “further investigation.”

4 Finally, although not necessary to prevail on this summary judgment motion, Juniper has
5 presented through its expert Dr. Alexander affirmative evidence that the SRX and J series products
6 do not use the Multiservices code. As explained above, Dr. Alexander conducted an analysis of
7 certain “make” and “manifest” files pertaining to the SRX and J series products, including the
8 “flow daemon” (“flowd”) relating to supposedly infringing stateful firewall functionality.
9 Alexander Ex. A ¶¶ 80-85; *see also* Hefazi Ex. 9 (9/20/2011 Narayanaswamy Depo. Tr.) at 99.
10 Dr. Alexander explained that Juniper uses the “makefiles” as part of a standard process to
11 “compile the source code building blocks for its products into executable object code.” Alexander
12 Ex. A ¶ 77. By analyzing these files, Dr. Alexander was able to determine that (for example) the
13 sole alleged “component” for which Dr. Nettles provides any source code analysis (the “CPCD”
14 plugin) is not used in the Accused Products. Alexander Ex. A ¶¶ 75-83. As another example, Dr.
15 Alexander confirmed that the “Service Set” concept (i.e., the starting point for Dr. Nettles’s flow-
16 based processing analysis, *see* Hefazi Ex. 8 at 14, 37) does not exist in the Accused Products.⁷
17 Not only has Implicit failed to meet its burden of proving the applicability of its cited code, but the
18 only evidence of record on this point (from Juniper’s expert Dr. Alexander) affirmatively
19 evidences that the code is not applicable to the SRX and J series products. Juniper has thus not
20 only met but exceeded the standard for summary judgment in this case.

21 **B. Implicit Cannot Show That the Multiservices Module is an Appropriate Proxy**
22 **for Analyzing Infringement by the SRX and J Series Products**

23 In an attempt to compensate for its failure to demonstrate infringement by direct analysis

24 _____
25 ⁶ Dr. Alexander pointed out in his report that Dr. Nettles could have, for example,
26 attempted to provide such proof through analysis of the “makefiles” that Juniper produced to
27 Implicit. Alexander Ex. A ¶¶ 77-79.

28 ⁷ Dr. Alexander analyzed the source code cited by Dr. Nettles relating to Service Sets and
demonstrates that it is designed for use with the Multiservices PIC. Alexander Ex. A ¶¶ 87-98; *see*
also Hefazi Ex. 4 (Treskunov Report) at 6 (diagram showing “service sets” in connection with
Multiservices module); Hefazi Ex. 7 at 190:17-22 (connecting concept of “service sets” with
Multiservices module); *see also* Hefazi Ex. 10 at 188:16 – 189:12 (regarding Treskunov report).

1 of the actual source code for the accused products, Implicit has turned to a pair of back-up
 2 arguments. First, Implicit argues that it is acceptable for it to prove infringement in this case by
 3 analyzing something other than the accused products, under a “proxy” theory of infringement.
 4 This theory at its core depends on an assertion from Dr. Nettles in his deposition (although never
 5 stated in his report) that “Juniper has basically represented that their products all work the same.”
 6 Hefazi Ex. 5 at 130:8-9. From this, Implicit reasons that by analyzing code for one Juniper
 7 product (the Multiservices module), Dr. Nettles has effectively established infringement by all
 8 Juniper products, thus obviating the need for specific analysis of the SRX and J series products.

9 Implicit’s “proxy” theory is fundamentally flawed both legally and factually. It is true that
 10 in some cases courts have found it acceptable to analyze infringement for a single representative
 11 accused product as a proxy for a set of related accused products that are all very similar. Indeed,
 12 where there is no genuine dispute about the similarity of the products at issue, parties can and do
 13 agree to identify a single accused product to serve as a representative proxy for purposes of the
 14 infringement analysis. Alternatively, even in the absence of an agreement between the parties, the
 15 patentee can elect to marshal and rely on its own evidence that the single accused product it
 16 analyzed is similar to—and therefore fairly representative of—each of a larger set of accused
 17 products. *See, e.g., Eugene Baratto, LLC v. Brushstrokes Fine Art, Inc.*, 701 F.Supp.2d 1068,
 18 1080-81 (W.D. Wis. March 24, 2010) (patentee “may group products” but “must support [its]
 19 grouping choices with evidence that every product . . . actually functions in the same manner.”)

20 What Implicit is seeking to do here, however, is significantly different from these standard
 21 “proxy” scenarios, because the proxy product implicitly chosen in this case—the Multiservices
 22 module as used in a Juniper M, MX, T, or TX router—***is not an accused product at all***. It would
 23 be one thing if Implicit had elected to analyze (for example) the SRX 100 product and extrapolate
 24 an infringement theory for the other SRX products in that family. But Implicit has not even done
 25 that. To the contrary, Dr. Nettles testified that did not even form any opinion that the Multiservices
 26 products infringe the patents-in-suit, although he had performed some analysis on that topic.
 27 Ex. 5 at 125:1-20, 128:8-25. There is no reason to think that Dr. Nettles should be able prove that
 28 the SRX and J series products infringe by analyzing and citing to evidence pertaining to a product

1 for which he has not even formed an opinion regarding infringement.⁸

2 In any event, Implicit has not come close to showing that the Multiservices products are, in
3 fact, an appropriate proxy for use in analyzing infringement by the accused SRX and J series
4 products. It is Implicit—not Juniper—that bears the burden here, as the case law makes clear:
5 “[Dr. Nettles] cannot simply ‘assume’ that all of [Juniper’s] products are like the one [Dr. Nettles]
6 tested and thereby shift to [Juniper] the burden to show that it is not the case.” *L&W, Inc. v.*
7 *Shertech, Inc.*, 471 F.3d 1311, 1318 (Fed. Cir. 2006).⁹ But Dr. Nettles admitted at his deposition
8 that his infringement analysis proceeded on the basis of just such an assumption:

9 Q. In your view why does it matter whether the facts under
10 consideration in this case pertained to one of the accused products or
one of the Juniper services routers?

11 A. Well, at -- at some level it doesn't. I mean, Juniper has basically
12 represented that their products all work the same. . . .

13 Hefazi Ex. 5 at 130:1-13; *see also id.* at 244:13-16.¹⁰

14 Dr. Nettles’s assumption—that “Juniper has basically represented that their products all
15 work the same”—is utterly unfounded. For one, Juniper never agreed in this case that the
16 Multiservices products “work the same” as the SRX and J series products, nor did the parties ever
17 reach any agreement that an analysis of the Multiservices products could serve as an acceptable
18 proxy for resolving the infringement issue as to the SRX and J series products. In fact, Juniper
19 repeatedly told Implicit just the opposite during discovery:

20 The implementation of JUNOS is not uniform across all [accused
21 products] and the JUNOS functionality that exists in one product
22 does not necessarily exist in others. As such, Implicit’s assumption
that all products operating using the JUNOS software implement the
same functionality is improper and incorrect.

24 ⁸ *See GE v. Joiner*, 522 U.S. 136 (1997) (where opinion evidence is “connected to existing
25 data only by the *ipse dixit* of the expert. . . [a] court may conclude that there is simply too great an
analytical gap between the data and the opinion preferred” and refuse to admit it).

26 ⁹ *See also Medtronics Vascular, Inc. v. Boston Scientific Corp.*, 2008 U.S. Dist. LEXIS
53373, *7 (E.D. Tex. July 11, 2008); *Alloc, Inc. v. Pergo, L.L.C.*, 2010 WL 3860382, *10 (E.D.
27 Wis. Sept. 30, 2010).

28 ¹⁰ These deposition statements improperly go beyond the opinions expressed in Dr. Nettles
report, where he stopped well short of making the claim that Juniper’s products “all work the
same.” Hefazi Ex. 8 ¶¶ 7-9.

1 Hefazi Ex. 11.¹¹ Thus, Implicit had plenty of notice that it was up to Implicit to prove up the
 2 appropriateness of a “proxy” approach, if it could.

3 But the factual record is also devoid of any affirmative evidence that could support an
 4 allegation that Juniper’s products “all work the same.” As explained above, Dr. Nettles never did
 5 any of his own analysis on the software code review computer, never compared the Multiservices
 6 code to the SRX or J series code to look for similarities or differences, and indeed failed to
 7 examine *any* source code for the SRX and J series products. Hefazi Ex. 5 at 100:4-11, 105:11-13.
 8 Nor did he ever ask Mr. Treskunov to do such an analysis, and even after hundreds of hours of
 9 code review, Mr. Treskunov “never communicated” to Dr. Nettles any supposed finding that “all
 10 of the Juniper accused products operated in essentially the same manner as the Multiservices
 11 module for purposes of infringement.” Hefazi Ex. 7 at 169:6-17.

12 Not surprisingly, the infringement report of Dr. Nettles contains no comparative analysis
 13 of source code or other internal Juniper technical documents that might demonstrate a similarity
 14 between the Multiservices products and the SRX and J series products. Instead, the Nettles report
 15 contains just three paragraphs that refer indirectly to this issue, relying solely on a 12-page
 16 “product category brochure” (a Juniper marketing document) regarding Juniper’s JUNOS
 17 operating system.¹² Hefazi Ex. 8 ¶¶ 7-9; Hefazi Ex. 13 (JUNOS Brochure). As this document
 18 explains, JUNOS is called a single operating system in the sense that various Juniper products can
 19 selectively pull different subsets of software code from a “single source code base.” Juniper’s
 20 engineers consistently confirmed this fact, explaining that a single operating system “does not
 21 mean that all of that software [from the single source code base] is loaded on each and every
 22 system.” Hefazi Ex.14 (6/19/2012 Tavakoli Tr.) at 151:4-19, 150:20-24. To the contrary,

23
 24 ¹¹ See also Hefazi Ex. 12 (“To use an analogy, versions of the Windows operating system
 25 can run on a phone, a laptop, a server, or even an automobile, but no one would say that these
 disparate products ‘actually function’ in the same way.”).

26 ¹² Dr. Nettles whole-hearted reliance on this marketing document on this critical point is
 27 questionable given he has elsewhere expressed the view that marketing pieces are “dubious,”
 28 “high level,” and “not very precise.” Hefazi Ex. 7 at 122:1-14 (“it’s sort of marketing material, so
 I’m always dubious about marketing material . . . marketing material is generally high level. It’s
 usually meant for a lay audience, so it’s often not very precise.”). Dr. Nettles also attempted to
 identify other support for this proposition in his report. Those attempts were unsuccessful. Hefazi
 Ex. 5 at 233:6-238:7.

1 “JUNOS is a very big and complex operating system with many differences between the different
 2 products.” Hefazi Ex. 15 at 19:3-9. “If you look at [Juniper’s] product portfolio, there’s many
 3 different things in there, and those work differently inside of the code base.” *Id.* And indeed, this
 4 is exactly what the brochure that Dr. Nettles relies upon confirms, expressly calling out the many
 5 *differences* between the various product lines, including the SRX, J, M, MX, and T series
 6 products. Hefazi Ex. 13 at 9-10.¹³

7 Thus, there is literally no evidence that Juniper’s products “all work the same” for
 8 purposes of the infringement analysis in this case. Dr. Nettles’s assumptions to the contrary are
 9 nothing more than unfounded speculation and therefore insufficient to survive summary judgment.

10 **C. Implicit’s Infringement Case Cannot Stand Without the Source Code**
 11 **“Analysis” in the Nettles Report**

12 Implicit’s second argument in response to its failure to do a direct analysis of the actual
 13 accused product source code is to argue that the flawed source code analysis was never that
 14 important to begin with, that this analysis should now be disregarded in its entirety, and that
 15 Implicit’s infringement case can proceed to trial based on its non-source-code evidence alone.

16 There are a number of problems with Implicit’s “whoops, never mind” argument. For one,
 17 it ignores the significant history in this case on the issue of source code, one that is familiar to the
 18 Court. Juniper specifically and repeatedly demanded during discovery that Implicit identify the
 19 source code citations that it contended supported its theories of infringement in this case. As the
 20 caselaw cited in Juniper’s discovery motions explained:

21 [A]fter a plaintiff-patentee has had a reasonable opportunity to
 22 review the source code for the defendant's accused software product,
 23 the patentee's time for trolling the proverbial waters for a theory of
 infringement comes to an end, and the patentee must fish or cut bait
 with respect to its specific theory of infringement

24 *Big Baboon Corp. v. Dell, Inc.*, 723 F. Supp. 2d 1224, 1229 (C.D. Cal. 2010). The Court

26 ¹³ Moreover, as Dr. Alexander explained, “the claims are very specific in the type of state
 27 information that must be used, as well as the nature of the storing and processing each component
 28 must perform for the different types of state information.” Alexander Ex. A ¶ 71-74. Thus, unless
 Implicit can show similarity between the products in handling of state information, etc., then the
 proof-by-proxy approach fails. *See Eugene Baratto*, 701 F.Supp.2d at 1080-81.

1 understood in granting Juniper's motions that what Juniper sought was *specificity* in Implicit's
 2 infringement contentions that would benefit Juniper in preparing its defense of this case; that the
 3 time to "fish" was over. Implicit thereafter claimed it had complied with these orders by
 4 identifying the source code citations that relate to the Multiservices module. Implicit never sought
 5 to amend these narrowed contentions, so these same pinpoint citations because the foundation of
 6 Dr. Nettles's expert report. It is simply too late for Implicit to now try to advance some new
 7 theory of infringement based on different source code; that ship has sailed.

8 Nor can Implicit seriously contend that Dr. Nettles's infringement theories based on the
 9 wrong source code can survive because the "non-source-code" evidence cited in the Nettles report
 10 is actually the "broadest and strongest support for infringement in this case." Hefazi Ex. 5 at
 11 144:7-14. The source code is the *best evidence* of exactly how the Juniper products were designed
 12 and functioned. In fact, as the Federal Circuit has recognized, sometimes "a review of source code
 13 may be the only way to make a comparison of an accused device to proposed patent claims."
 14 *Scanner Techs. Corp. v. ICOS Vision Corp.*, 528 F.3d 1365, 1376 (Fed. Cir. 2008).¹⁴ It is true
 15 that, after reviewing the Alexander report, Dr. Nettles started expounding the supposed
 16 "limitation[s] of source code analysis." *Id.* at 146:24-25.¹⁵ But the Nettles report tells a different
 17 story: in the portion explaining how Juniper allegedly satisfies each of the limitations of the
 18 asserted claims, Dr. Nettles relies almost exclusively on the code for the details of his analysis.
 19 And when asked whether there was any "caveat or caution against reliance on source code in [his]
 20 report," Dr. Nettles admitted that "there isn't such a caveat." Hefazi Ex. 5 at 155:21-157:4.

21 In any event, it is clear that the non-source-code evidence cited in the Nettles report is,
 22 standing alone, inadequate to survive summary judgment. Setting aside the source code evidence,
 23

24 ¹⁴ Indeed, key concepts and functionality in Implicit's infringement analysis, including that
 25 of the "Service Set," Redacted and retrieval of component-specific "state information" on
 a packet-by-packet basis are never discussed in the documentation for the accused products cited
 by Dr. Nettles.

26 ¹⁵ For example, Dr. Nettles suggested that in some cases it was "mathematically
 27 impossible" to prove infringement through "static" source code review. Ex. 5 at 147:10-12. But
 28 he also said he had no opinion as to whether the claims of the patents-in-suit presented such a
 case, *id.* at 147:19 –148:6, and in any event he admitted that he never inquired of Mr. Treskunov
 about performing a non-static analysis. *Id.* at 148:7-12.

Implicit is left with nothing more than conclusory statements and a series of unexplained block quotations that at best describe the functionality of the SRX and J series products at a high level, but do not provide evidence of how the accused products allegedly practice each limitation of the asserted claims—which is the heart of an infringement analysis. Dr. Nettles simply does not map the limitations of the claims to the accused products by any means other than source code. Without the source code, there is nothing in Dr. Nettles report that can “pinpoint where [the claim] elements are found in the accused devices.” *Pixion, Inc. v. Citrix Systems, Inc.*, 2012 WL 3313533 (N.D. Cal. Aug. 13, 2012) (Illston, J.); *Novartis*, 271 F.3d at 1054 (“Even if we were to accept without question that Dr. [Nettles] began with the proper parameters, we are left completely in the dark as to how he employed them.”).

Implicit’s failure of proof may be best demonstrated by examining Dr. Nettles’s report itself. A good example is his discussion of element 1f of claim 1 of the ‘163 patent. The only specific non-source-code “analysis” here is a block quote from Juniper’s “IDP Series” Guide:

Evidence ‘163 C1 1f(2)

When the IDP engine processes security policy rules, it examines the session, beginning with the first packet, to identify a match. To match service or application, the IDP engine first compares the session against the application identification cache to identify the application. If the session does not match the application identification cache, the IDP engine processes the session against the application signatures. If the IDP engine is still unable to determine the application, it uses the standard application protocol and port.

Source: *IDP Series Concepts and Examples Guide*, Juniper Networks, Published Feb. 2011, http://www.juniper.net/techpubs/en_US/idp5.1/information-products/topic-collections/idp-5-1-r1-concepts-examples.pdf, page 96

126. Thus, in my opinion and as shown above and throughout this report, element 1f is satisfied by JN1’s accused products.

Hefazi Ex. 8 ¶ 125; Hefazi Ex. 16 (IDP Series Guide) at 96.¹⁶ This high-level “analysis” is deficient on its face because Dr. Nettles provides no explanation of how this block quote purportedly establishes the relevant claim limitation—e.g., what is the “component” or the corresponding “processing,” what is the “state information” and how is it “retrieved” and “used” in the “processing,” and so forth. *Pixion*, 2012 WL 3313533, *7 (must “supply at a minimum

¹⁶ Implicit never cited any source code pertaining to this “application identification” feature, although ordered to do so during the discovery phase of the case. It appears that Mr. Treskunov was not familiar with the “application identification” functionality and had not identified any specific source code citations for this feature. Hefazi Ex. 25 at IMP141520.

some description about the specific features that show infringement” to survive summary judgment).¹⁷ Moreover, the cited evidence for element 1f is further deficient because it originates from a document that itself states it is describing the IDP series products, and not the SRX or J series products. Alexander Ex. A ¶ 98.¹⁸ As with the Multiservices products, Dr. Nettles provides no explanation of why or how the functionality of this non-accused product can be relevant to infringement in this case. *Pixion*, 2012 WL 3313533, at *7.

Thus, upon removing the sole two pieces of evidence cited for element 1f—both of which correspond to products other than the accused products—Dr. Nettles is left with no evidence or analysis to support this claim element. Of course, a finding of infringement requires that “*every claim limitation* is found in the accused device,” *id.* at *6, and so this failure on element 1f is sufficient to merit summary judgment of non-infringement for the entirety of claim 1. And because Dr. Nettles goes on to incorporate by reference the flawed element 1f “analysis” with respect to at least one element for every other asserted claim in this case, this failure of proof undermines Implicit’s infringement case for every other asserted claim in this case, as well.¹⁹

III. IN THE ALTERNATIVE, IMPLICIT’S INFRINGEMENT CLAIMS FAIL EVEN IF ALL OF DR. NETTLES’S CITED EVIDENCE IS DEEMED APPLICABLE TO THE ACCUSED PRODUCTS, AS HE STILL FAILS TO SATISFY AT LEAST THREE ELEMENTS OF THE ASSERTED CLAIMS.

Even assuming for the sake of argument that all of the evidence cited by Dr. Nettles applies to the actual accused products in this case (although it clearly does not), summary judgment of non-infringement is still appropriate because there are at least three elements of each of the asserted claims for which Implicit has presented no evidence or argument sufficient to permit a reasonable jury to find infringement of those claims at trial. As “the expert report. . . defines the metes and bounds of an expert’s trial testimony,” *In re Oracle*, 2009 U.S. Dist. LEXIS

¹⁷ Dr. Nettles later testified he did not remember the “application identification” feature of the accused products “being a -- particularly important to my analysis.” Ex. 5 at 149:6-23.

¹⁸ The cited document explains that Juniper provides “intrusion detection and prevention (IDP) technology” in at least the IDP, ISG, and SRX families of products, but that “[t]his guide describes the IDP Series appliances.” Hefazi Ex. 16 (IDP Series Guide) at 3. Note that Implicit previously accused the IDP series of products of infringement in this case but later dropped them.

¹⁹ See Hefazi Ex. 8 ¶¶ 178, 231, 288, 336, 390 (‘163 patent, element 15d & 35d, and ‘857 patent, element 1e, 4d, and 10e).

50995, *84 (N.D. Cal. June 16, 2009) (Illston, J.), these deficiencies in the Nettles report provide independent bases upon which summary judgment of non-infringement can and should be granted. *Pixion*, 2012 WL 3313533, at *7. These three claim elements are taken up in turn below.

A. Implicit Cannot Show That The Accused Products Satisfy The “Selecting Individual Components” Limitation

Each of the asserted claims includes the limitation “selecting individual components.” The Court has construed “selecting individual components” as “selecting the individual software routines of the sequence *so that the input and output format of the software routines are compatible.*” Hefazi Ex. 1 at 11 (emphasis added). In construing the term, the Court explained that “a *necessary* part of selecting individual components is *determining the compatibility between the output of one software routine and the input of the next.*” *Id.* (emphasis added). The parties have agreed that this “determining” aspect is indeed part of “selecting individual components,” as explained in the Court’s order.²⁰

Thus, in order to show that the “selecting individual components” limitation is satisfied, Dr. Nettles must point to some functionality in the SRX and J series products that determines whether the input and output formats of the selected software routines are indeed compatible. As Juniper’s products do not do this, Dr. Nettles was unable to find any evidence that they do. To the contrary, Dr. Nettles’s report contains no evidence, explanation, or identification of any purported compatibility checking in the SRX and J series products that would satisfy the “selecting individual components” limitation. At most, Dr. Nettles alleges at one point of his report that “[t]he accused products provide components that operate on the data in sequence with the output of one component being the input of the next.” Hefazi Ex. 8 ¶ 72. But he provides no explanation of how this supposedly occurs, much less evidence to support his conclusory statement. And the statement pointedly avoids any mention of “compatibility” in any event.

²⁰ Specifically, INI’s corporate representative and Dr. Nettles have both confirmed agreement with the Court’s statement. *See* Hefazi Ex. 19 (Balassanian Depo.) at 1001:1-8, 1002:1-2; Hefazi Ex. 5 (10/09/2012 Nettles Depo.) at 251:16-253:1; *see also* Hefazi Ex. 20 (7/10/2012 INI’s Response in the *inter partes* ‘857 Reexam) at 29-30; Ex. 21 (6/4/2012 Implicit Response to Office Action) at 20; Hefazi Ex. 22 at Interrogatory No. 18.

1 Dr. Nettles was also unable to identify any such compatibility checking in the SRX and J
 2 series products during his deposition. *See, e.g.*, Hefazi Ex. 5 at 252:2-253:3. For example,
 3 Dr. Nettles was asked to identify at what point in time he believed the alleged compatibility
 4 checking occurred in the accused systems. The only response he could muster after repeated
 5 questioning was that the accused products work basically the same as something called “label map
 6 get” (which is part of an embodiment described in the specification of the patents-in-suit):

7 Q. Is it your opinion that Juniper does the compatibility checking
 before or after the first packet?

8 A. It's my understanding that Juniper system works in the same
 way as label map get.

9 Q. Is it before or after? We have to know. We're going into trial
 10 and we're going to have Summary Judgment. Let's hear it. Is it
 before or after, in the Juniper products?

11 A. Again, it's my understanding that the Court has made it clear
 that label map get satisfies this particular requirement. And that
 12 Juniper's products work in the same manner as label map get.

13 Q. And you can't tell me sitting here today whether this
 compatibility matching in the Juniper products happens before or
 14 after the first packet; correct?

15 A. I've –

16 Q. Will you tell me?

17 A. I've given you an answer to your question, which is my
 understanding is that the Court has been very clear that label map
 18 get satisfies the requirements of this particular claim term. And my
 understanding is that -- is that Juniper's products work in the same
 manner as label map get.

19 Hefazi Ex. 5 at 252:2-258:25. Oddly, despite this testimony, Dr. Nettles's report actually contains
 20 no discussion or analysis whatsoever comparing “label map get” with the accused Juniper
 21 products. In fact, “label map get” is not even mentioned anywhere in Dr. Nettles's report. Thus,
 22 Dr. Nettles's conclusory statement at his deposition cannot support a finding that this element is
 23 met by the SRX and J series products.²¹

24 Dr. Nettles was also asked to identify “the best piece of evidence you have cited in your
 25 report for the proposition that Juniper performs this input/output matching limitation.” Hefazi

26
 27 ²¹ Such an analytical approach would have been improper in any event, as infringement
 28 requires comparing the accused product against the “properly construed claims” and not against “a
 preferred embodiment described in the patent.” *Atlantic Thermoplastics Co. v. Faytex Corp.*, 974
 F.2d 1299 (Fed. Cir. 1994).

Ex. 5 at 257:21-259:10. In responding to this question, Dr. Nettles did not turn to the element-by-element section in his technical analysis, or indeed any of the technical section of his report. Instead, he turned to the “Value Section” of his report (relating to damages) and pointed to an irrelevant citation from a deposition of a technical Rule 30(b)(6) witness for Juniper (Mr. Krishna Narayanaswamy). *Id.*²² Dr. Nettles explained that although the *cited* portion of the deposition was not relevant on this point, there was other, *uncited* material from the deposition regarding “configuration information” that might satisfy the claim limitation (though he did not remember where). *Id.* Given that Dr. Nettles never cited any such evidence in his expert report in connection with this claim element—indeed, the technical section of his report never cites to Mr. Narayanaswamy at all—this is likewise insufficient to survive summary judgment.²³

In sum, because Dr. Nettles fails to address how the SRX and J series products “select[] the individual software routines of the sequence so that the input and output format of the software routines are compatible,” no reasonable jury could find that Implicit satisfied its burden as to the “selecting individual components” limitation.

B. Implicit Cannot Show That The Accused Products Include A “Plurality of Components” Satisfying The Claims, Including The “State Information” Limitations

The asserted claims as construed require a sequence of multiple “components” or “software routines” (*see* Hefazi Ex. 1 at 5), each of which must satisfy a number of specific requirements set forth in the claims. For example, each component must perform specific steps involving “state information,” which the Court has construed to mean “information *specific to a software routine* for a specific message that is not information related to an overall path.” Hefazi *Id.* at 14. Claim 1 of the ‘163 patent is representative:

“... for each of a plurality of packets of the message in sequence, for *each* of a *plurality of components* in the identified non-predefined sequence,

[1] *retrieving state information relating to* performing the processing of the component with *the previous packet* of the message;

²² *See* Hefazi Ex. 27 (Nettles Report) at 35; Hefazi Ex. 9 (Narayanaswamy) at 123:21-24.

²³ In any event, the Narayanaswamy deposition contains no indication that the SRX and J series products perform a compatibility determination per the asserted claims.

[2] *performing the processing* of the identified component *with the* packet and the *retrieved state information*; and

[3] *storing state information* relating to the processing of the component with packet *for use when processing the next packet* of the message.”

Each of these “state information” limitations must be satisfied for “each of a plurality of packets” *and* “each of a plurality of components,” as Dr. Nettles concedes. Hefazi Ex. 5 at 226:2-6. Yet Dr. Nettles’s report fails to disclose any evidence, explanation, or identification of a “plurality of components” in the SRX and J series products such that each “component” satisfies the specific “state information” limitations of the claims.

As an initial matter, although Dr. Nettles apparently identifies a number of accused “components” in Exhibit 3 to his report (i.e., a list of purported Multiservices “plugins”) (Hefazi Ex. 23), he only attempts an infringement analysis for *one* of these: the so-called “CPCD plugin.”²⁴ He never analyzes any software code for the other plugins in Exhibit 3 and does not even identify what are the alleged “software routines” corresponding to these plugins. *Id.*²⁵ This lack of analysis is perhaps unsurprising as Dr. Nettles admitted in deposition that the Exhibit 3 list of plugins was something that he copied directly from the Treskunov report without even asking how it was generated. Hefazi Ex. 5 at 185:8-186:3; *see also* Hefazi Ex. 4 (Treskunov Report) at 25. Merely listing purported plugin names is wholly insufficient to show that each of the accused “components” satisfies the specific “state information” limitations of the asserted claims.²⁶

During his deposition, Dr. Nettles also claimed to have disclosed in his report other

²⁴ When Juniper pointed out during discovery that Implicit had failed to conduct an analysis for any of the Exhibit 3 plugins other than CPCD, INI’s own software code consultant candidly admitted that this was “somewhat a true statement.” Hefazi Ex. 25 at IMP141521.

²⁵ For example, during his deposition, Dr. Nettles identified “junos-nat” as an exemplary component from the Exhibit 3 list in addition to CPCD (Hefazi Ex. 5 at 177:16-23), but then admitted that his report contained no evidence that “junos-nat” was actually a software routine (*id.* at 186:13 – 187:19). Dr. Nettles could not say where in the Juniper code junos-nat” was implemented (*id.* at 183:6 – 185:3) and admitted that no such code was ever identified in his report (*id.* at 185:4-7).

²⁶ For the same reason, Dr. Nettles cannot demonstrate that each of the accused components performs “converting” (see ‘163 patent, claim 1). In fact, Dr. Nettles admitted that the single plugin he did analyze, CPCD, did not perform conversion. Hefazi Ex. 5 at 196:10-11 (“cpd doesn’t do such conversion.”). This provides an additional reason to grant summary judgment as to claim 1 of the ‘163 patent and possibly other claims (should the Court elect to find Implicit has disclaimed non-converting components in the reexamination proceedings).

1 examples of “components” that satisfied the claims in addition to those listed in Exhibit 3. Hefazi
 2 Ex. 5 at 189:5-9. But Dr. Nettles was unable to identify any such components that corresponded
 3 to the “state information” elements in his report, claiming instead he was not obligated to include
 4 such identification and analysis in his report. *Id.* at 201:16-202:21. (“I don’t have to come and
 5 say, look, . . . this is a component, this is a component, this is a component.”). After repeated
 6 further questioning, Dr. Nettles did finally state that he thought “SSL” was “clearly a component,”
 7 but then immediately admitted that he did not actually disclose “in the report here I’m identifying
 8 SSL in part to say that it’s a component.” *Id.* at 203:7-205:23. And when asked where was the
 9 support in his report for the “state information” limitations (in particular element 1g), Dr. Nettles
 10 admitted that “I probably don’t have an explicit place in my report where I say SSL reads and
 11 writes state.” *Id.* at 210:7-211:9; *see also id.* at 212:6-24, 232:24-233:1).²⁷ Of course, Dr. Nettles
 12 cannot defeat summary judgment by attempting to supplement his report with additional,
 13 conclusory deposition testimony. *In re Oracle*, 2009 U.S. Dist. LEXIS 50995, *84.

14 As another example, Dr. Nettles testified during his deposition that a figure reproduced in
 15 the “Overview” section of his report demonstrated that “state information stored for one packet is
 16 then used for a subsequent packet” per the claims. Hefazi Ex. 5 at 164:17-20; Hefazi Ex. 8
 17 (Nettles Report) at 8. But that figure (a high-level illustration of a flow module) does not even
 18 come close to evidencing or explaining the very specific steps required by the claims, and the
 19 Nettles report never identifies (for example) what is the state information, how it relates to
 20 processing of packets, or how it is stored for use when processing the next packet of the message.
 21 This type of “opaque identification is not enough to permit any reasonable juror to make the leap”
 22 in finding infringement. *Pixion*, 2012 WL 3313533, at *7.

23 Finally, Dr. Nettles fails to satisfy the “state information” limitations even for the one
 24 CPCD plugin for which he did actually review the software code. Dr. Nettles provides no
 25 explanation or analysis of the cited code excerpts for CPCD other than a few cryptic notes that do
 26

27 ²⁷ As another example, Dr. Nettles said during his deposition that “NAT logging” might
 28 satisfy the “state information” elements for a NAT component, but then admitted that he never
 pointed to “NAT logging” anywhere in his report. Ex. 5 at 228:2-18.

1 not map clearly to the language of the claim elements. For example, for the element 1g “storing”
 2 step, Dr. Nettles provides pinpoint citations to the **Redacted**, which he claims
 3 “is called while processing the first packet of the session.” Hefazi Ex. 8 at 66-67. But of course,
 4 the claims themselves require the storing step not just for the “first packet” but for “each of a
 5 plurality of packets” passing through the CPCD plugin, so as to ensure that the state information is
 6 properly stored “for use when processing the next packet of the message” (to use the language of
 7 the claims). Dr. Nettles simply fails to address this packet-by-packet aspect of the claims in his
 8 infringement analysis. Similarly, Dr. Nettles states that the information stored in the cited code is
 9 “opaque (plugin dependent)” (*id.* at 66) but never offers any reason or analysis why this is so. In
 10 order to demonstrate that this is “not information related to an overall path” (as required by the
 11 Court’s construction), there would need to be some evidence that the information is used just for
 12 this plugin, and not for all the other plugins in a particular sequence of components. Alexander
 13 Ex. A (Alexander Report) ¶ 213. Dr. Nettles fails to provide such evidence.

14 Because Dr. Nettles fails to address how the SRX and J series products include multiple
 15 “components,” each of which performs the specific “state information” limitations of the claims,
 16 no reasonable jury could find that Implicit satisfied its burden with respect to these limitations.

17 **C. Implicit Cannot Show That The Accused Products Perform The Step Of**
 18 **“Dynamically Identifying . . . After The First Packet.”**

19 Finally, each of the asserted claims requires the limitation of “dynamically identifying” a
 20 “sequence of components.” The claims and the Court’s construction make clear that “identifying”
 21 a sequence must occur *after* the first packet of the message is received; it is not enough (as
 22 Implicit proposed during claim construction) that the step merely occur during “runtime.” Hefazi
 23 Ex. 1 at 5-7. Yet Dr. Nettles’s report fails to disclose any evidence, explanation, or identification
 24 of the necessary aspect of “dynamically identifying . . . [a] sequence of components” in the SRX
 25 and J series products in the manner required by the claims (i.e., after the first packet of a message).

26 Initially, Implicit’s own analysis confirms that the accused sequence of components (a list
 27 of plugins) is something that exists *before* the first packet, contrary to the requirements of the
 28

1 asserted claims. Specifically, Dr. Nettles states that “an ordered list of the Plugins to be executed
 2 is obtained” from something called a “Service Set,” which comprises settings that an administrator
 3 has provided *before* the arrival of the first packet. Hefazi Ex. 8 at 19, 37; Alexander Ex. A at
 4 161-163. Implicit’s source code consultant Mr. Treskunov was even more blunt on this point in
 5 his report, admitting that “*the sequence of plugins to be applied to the first packet is known*
 6 *before the packet arrives.*” Hefazi Ex. 26 at IMP141522. In other words, upon receiving the first
 7 packet of a message, the accused products simply look up the predefined “ordered list of plugins”
 8 and then process each of the packets through this sequence of components. Hefazi Ex. 8 ¶ 42-43.
 9 These products do not wait until after the first packet is received to define the sequence or
 10 components, as required by the asserted claims.

11 Dr. Nettles advances two arguments in an attempt to show infringement, even though the
 12 accused products do not practice the “dynamically identifying” limitation. First, he suggests that
 13 this claim limitation can be met because when the first packet of the message is received, the
 14 accused products allegedly “instantiate” or copy the “ordered list of plugins” into a different
 15 location in memory. *Id.* ¶¶ 11, 35, 72. But the Court already rejected this attempt to re-write the
 16 claims during claim construction. Specifically, Implicit argued during the claim construction
 17 phase that the claim terms “create” and “form” (which are part of “dynamically identifying”) are
 18 satisfied by mere “instantiation” in memory. Hefazi Ex. 1 at 11. But the Court did not adopt this
 19 proposed construction after Juniper explained that there was no need to, as the claims themselves
 20 explain how “creating” or “forming” occurs (namely, by “selecting individual components”).
 21 Hefazi Ex. 24 (Markman Hearing Tr.) at 114:6-115:12. And of course, merely copying or
 22 “instantiating” into memory a predefined “ordered list of plugins” cannot constitute “dynamically
 23 identifying” a “sequence of components.” Deciding which components to use in a sequence (and
 24 in which order) must occur *after* the first packet of a message—precisely the point of the patents-
 25 in-suit and confirmed by the Court in the Markman Order.

26 This leaves Dr. Nettles’s second argument, which points to the source code and alleges that
 27 although the predefined “ordered list of plug-ins” is known before the first packet, the accused
 28

1 products may be able to “exclude” a certain plugin in that sequence “from future processing of
 2 packets of the current session” if the plugin issues a Redacted
 3 Hefazi Ex. 8 ¶ 85. This theory also fails. To begin, the claims make clear that “dynamically
 4 identifying includes *selecting* individual components after the first packet is received,” not
 5 *excluding* components. In this sense, Dr. Nettles’s theory of removing plugins from a predefined
 6 list is the logical opposite of dynamically selecting individual components. Indeed, Implicit’s
 7 source code consultant, Mr. Treskunov, recognized this point and concluded that the
 8 Redacted was “not relevant” because “it addresses the fact of remembering the plugins that
 9 should be *excluded* from future involvement in packet processing, but the claim. . . is about
 10 *preservation* of the sequence within session object.” Hefazi Ex. 26 at IMP141523.²⁸

11 Moreover, as Juniper’s expert Dr. Alexander has explained, the Redacted is
 12 merely invoked in cases of an error condition. Alexander Ex. A (Expert Report of Peter
 13 Alexander) at ¶¶ 166-171. In other words, plugins are excluded only when a user misconfigures
 14 the system or some other serious operational error condition is detected. Not only do the claims
 15 not cover a method for excluding error conditions, but Implicit also presents no evidence that the
 16 Redacted error condition is ever triggered. When asked whether he was “aware of any
 17 implementation of a Juniper product where the Redacted was invoked,” Dr. Nettles admitted
 18 he could not recall seeing any such implementation. Hefazi Ex. 5 at 249:23-259:23. And
 19 certainly none is disclosed in his report.

20 Finally, there is an internal inconsistency in Implicit’s infringement theory as it relates to
 21 the Redacted . As explained above, the CPCD plugin is the only plugin for which
 22 Implicit even attempts to show through an analysis of the actual code that the “state information”
 23 limitations are met. But Implicit simultaneously argues that the way the Juniper accused products
 24 dynamically create a new sequence after the first packet is received is by electing to *skip* the
 25 CPCD function. If the CPCD plugin is indeed skipped in the allegedly infringing mode, there is
 26

27 ²⁸ Furthermore, even if a certain plugin in a sequence were skipped, that would not in any
 28 way change the predefined order (i.e., sequence) of the remaining components. Thus, even if a
 Redacted were triggered and a plugin skipped, the remaining components would still be
 processed in the same predefined order that they were in before the first packet was received.

no way that the plugin could retrieve, use, or store “state information” as alleged by Dr. Nettles. Thus, Implicit’s final theory falls apart upon scrutiny as well.

In summary, because Dr. Nettles fails to demonstrate that the SRX and J series products satisfy the limitation of “dynamically identifying” a “sequence of components” at a point in time “after the first packet” (as required by the claims), no reasonable jury could find that Implicit satisfied its burden with respect to this limitation.

IV. IMPLICIT HAS FAILED TO DEMONSTRATE ANY ACTUAL INFRINGEMENT IN THE ACCUSED MANNER OR CONFIGURATION.

Finally, even if it had shown a theoretical capability of using or programming the SRX and J series products in an infringing manner (which it has not), Implicit has still failed to show the necessary predicate of actual infringing use of these products. *See, e.g., Acco*, 501 F.3d at 1313 (“A patentee must either point to specific instances of direct infringement or show that the accused device necessarily infringes the patent in suit.”). Implicit has accused fourteen different Juniper products (*see* Ex. 27 ¶ 70), each with different features and operating with countless different revisions of the accused software, yet Implicit fails to show “specific instances” of infringement in any particular, real-life circumstances, as required under Federal Circuit law. *See Fujitsu Ltd. v. LG Electronics, Inc.*, 620 F.3d, 1321, 1329 (Fed. Cir. 2010). For example, Implicit does not show which products for which customers (1) are operated in the concededly non-infringing “packet mode” (*see* Hefazi Ex. 8 ¶ 74); (2) use software revisions other than 11.1R2.3 (which is the only version that Mr. Treskunov analyzed, *see* Ex. 4 at 2); or (3) do not utilize stateful plugins, CPCD, service sets, or Redacted (*see, e.g.,* Hefazi Ex. 5 at 248:9-11 (“My understating. . . is that we don’t really know very much at all about what customers use or don’t use.”); *see also id.* at 248:23-25, 249:13-16, 250:12-19). Even under Implicit’s own theories, acts falling within any of these three categories cannot infringe the patents-in-suit. Thus, at a minimum, summary judgment of non-infringement is appropriate for alleged acts falling within any of these categories.

CONCLUSION

For the reasons set forth above, the Court should grant Juniper’s motion.

1 Dated: November 9, 2012

Respectfully submitted,

2
3 IRELL & MANELLA LLP

4 By: /s/ Nima Hefazi

5 David C. McPhie (dmcphie@irell.com)
6 Douglas J. Dixon (ddixon@irell.com)
7 840 Newport Center Drive, Suite 400
8 Newport Beach, California 92660-6324
9 Telephone: (949) 760-0991
10 Facsimile: (949) 760-5200

11 Morgan Chu (mchu@irell.com)
12 Jonathan S. Kagan (jkagan@irell.com)
13 1800 Avenue of the Stars, Suite 900
14 Los Angeles, California 90067-4276
15 Telephone: (310) 277-1010
16 Facsimile: (310) 203-7199

17 *Attorneys for Defendant*
18 *Juniper Networks, Inc.*
19
20
21
22
23
24
25
26
27
28